

FIG. 1

	<div><div>D UU</div><div>LL</div></div>	<div>U</div> <div>U1U2U3</div>		
	L1	C1		
	L2	C2		
	L3	C3		

FIG. 2

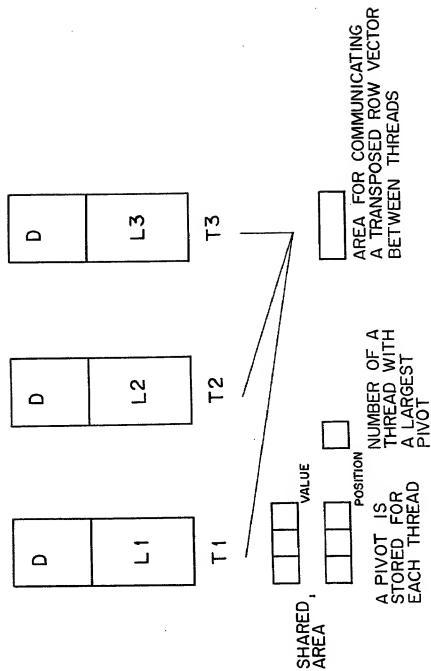


FIG. 3

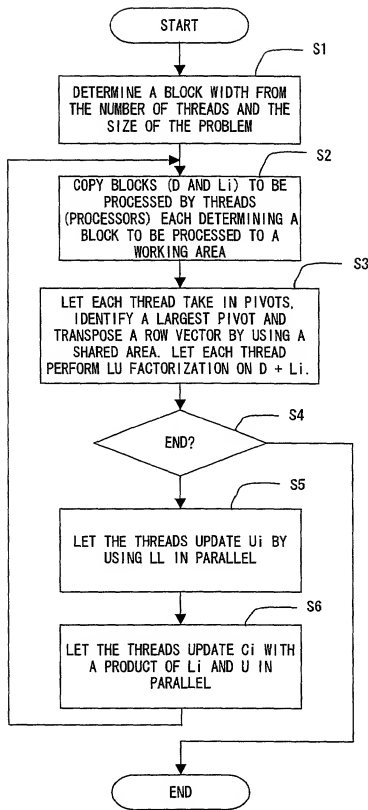


FIG. 4

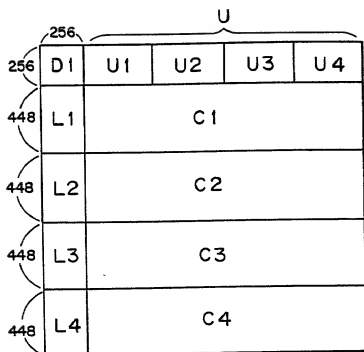


FIG. 5

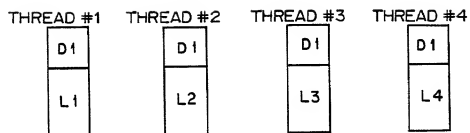
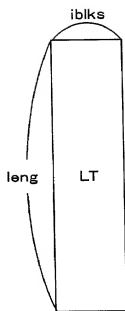


FIG. 6



```

DO i=1, iblks
  TMP=0.0 DO:jj=0
DO j=i, leng
  IF (ABS LT(j, i)), GT , TMP) THEN
    TMP=ABS(LT(j, i))
    jj=j
  ENDDO
ENDDO

```

(1)

```

IF (jj, GT, i) THEN
  DO k=1, iblks
    TMPX=LT(i, k)
    LT(i, k)=LT(jj, k)
    LT(jj, k)=TMPX
  ENDDO
END IF

```

(2)

```

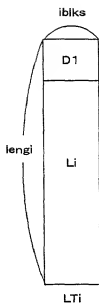
DO k=i+1, iblks
  LT(i, k)=LT(i, k) / LT(i, i)
ENDDO

DO k=i+1, iblks
DO l=i+1, leng
  LT(l, k)=LT(l, i) - LT(l, i) x LT(i, k)
ENDDO
ENDDO

```

(3)

FIG. 7



```

DO j=1, iblks
  TMP=0, 0 DO jj=0
DO j=1, lengi
  IF (ABS LTI(j, i), GT, TMP) THEN
    TMP=ABS(LTI(j, i))
    jj=i
  ENDDIF
ENDDO

```

(4)

```

pivpot( #THREAD )=jj
( #THREAD IS A THREAD NUMBER. IN THE
CASE OF PARALLEL PROCESSING BY 4
THREADS, #THREAD IS PRESCRIBED AS
1,2,3 AND 4 )

```

(5)

BARRIER SYNCHRONIZATION

```

IF( #THREAD, EQ, 1 )
  jx=0; GPIVOT=0
DO ix=1, 4
  IF (pivot(ix), GT, jx, AND, PIVOT(ix), GT, iblks) GPIVOT=ix
    (THE NUMBER OF A THREAD HAVING A LARGEST NUMBER)
ENDDO
END IF

```

(6)

BARRIER SYNCHRONIZATION

```

IF( #THREAD, EQ, GPIVOT ) THEN
  IF (jj, GT, i) THEN
    DO ix=1, iblks
      ROW(ix)=LTI(jj, ix)
    ENDDO
  END IF

```

(7)

BARRIER SYNCHRONIZATION

```

IF( GPIVOT, EQ, 0 ) THEN
  IF (jj, GT, i) THEN
    DO j=1, iblks,
      TMPW=LTI(i, ix)
      LTI(i, ix)=LTI(jj, ix)
      LTI(jj, ix)=TMPW
    ENDDO
  END IF
ELSE
  IF( #THREAD, EQ, GPIVOT ) THEN
    DO ix=1, iblks
      LTI(jj, ix)=LTI(i, ix)
      LTI(i, ix)=ROW(ix)
    ENDDO
  ELSE
    DO ix=1, iblks
      LTI(i, ix)=ROW(ix)
    ENDDO
  ENDIF

```

(8)

SINCE TRASPOSITION HAS
BEEN CARRIED OUT IN AN IP,
THE THREADS EXECUTE THE
PROCESSING IN PARALLEL

```

DO k=i+1, iblks,
  LTI(i, k)=LTI(i, k)/LTI(i, i)
ENDDO

```

(9)

```

DO k=i+1, iblks
DO l=i+1, lengi
  LTI(l, k)=LTI(l, k)-LTI(l, i) × LTI(i, k)
ENDDO
ENDDO

```

(10)

FIG. 8

ENDDO

256	D1	U1	U2	U3	U4
384	L1	C1			
384	L2	C2			
384	L3	C3			
384	L4	C4			

FIG. 9

```

subroutine LU(LTi, k, iblks, ist, nwid)
  (WHERE LTi IS USED BY THREADS FOR STORING (D1+Li),
   k IS THE SIZE OF THE FIRST ONE DIMENSION OF LTi,
   iblks IS THE BLOCK WIDTH,
   ist IS A POSITION TO START THE Lu FACTORIZATION AND
   nwid IS THE WIDTH OF AN OBJECT SUBJECTED TO THE Lu FACTORIZATION)
  IF (nwid, eq, 8), Then (A WIDTH OF 8 IS A MINIMUM).

```

```

  LTi(ist:k, ist, ist+nwid-1) IS SUBJECTED TO THE LU FACTORIZATION IN
  PARALLEL.

```

```

  [ HERE, THE PARTS (4) TO (10) OF FIG.9 ARE EXECUTED.
    IN THIS CASE, THE ROW-TRANSPOSING UNIT TRANSPPOSES
    LTi(i, 1, iblks) AT THE LENGTH iblk. ]

```

```

  else
    call LU(LTi, k, iblks, ist, nwid/2)
    call TRS( )
    UPDATE LTi(ist:ist+nwid/2-1, ist+nwid/2:ist+nwid). BY USING A
    LOWER-TRIANGULAR MATRIX LL OF LTi(ist:ist+nwid/2-1, ist:ist+nwid/2
    -1), UPDATE IT BY MULTIPLYING IT BY LL+ FROM THE LEFT.

```

```

  [ call MM( )
    LTi(ist+nwid/2:k, ist+nwid/2:ist+nwid)
    =LTi(ist+nwid/2:k, ist+nwid/2:ist+nwid)
    -LTi(ist+nwid/2:k, ist:ist+nwid/2-1) x
    LTi(ist:ist+nwid/2-1, ist+nwid/2:ist+nwid) ]

```

Barrier SYNCHRONIZATION

```

call LU(LTi, k, iblks, ist+nwid/2, nwid/2
end if
return
end subroutine

```

FIG. 10

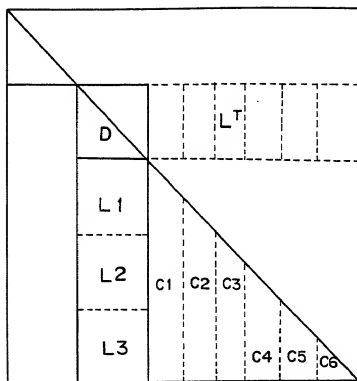


FIG. 11

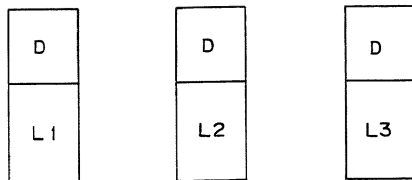


FIG. 12

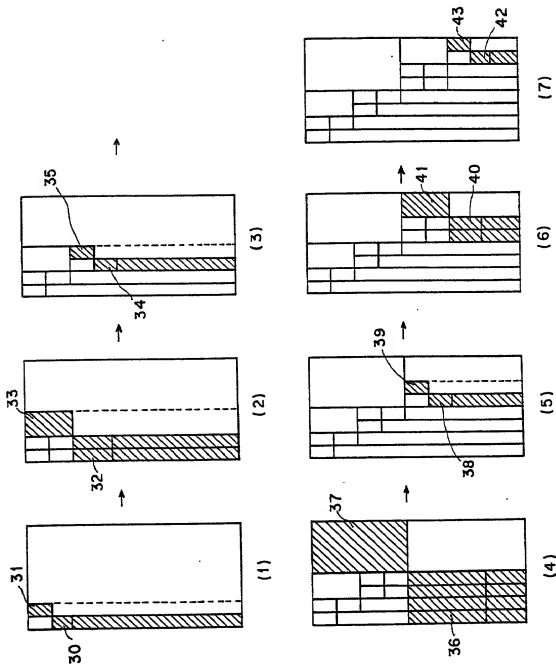


FIG. 13

The diagram illustrates a triangular matrix structure for a 4x4 system. The matrix is divided into a diagonal band of width $D1$ and a lower triangular region of width L . The diagonal band contains the main diagonal (DX) and three diagonal elements (DL). The lower triangular region contains four rows of elements ($L1, L2, L3, L4$), each with up to four elements ($C1, C2, C3, C4, C5, C6, C7, C8$).

FIG. 14

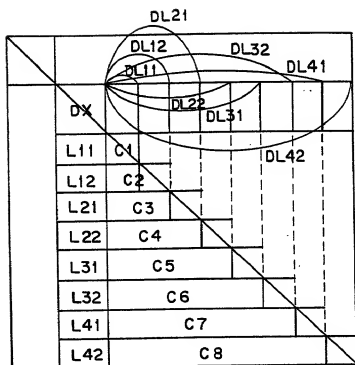


FIG. 15

```

subroutine LTD(LTi, k, iblks, ist, nwid)
  IF(nwid, EQ, 8) THEN (THE WIDTH OF 8 IS THE MINIMUM)
    DOi=ist, ist+7
    DOj=i+1, ist+7
    LTi(i, j)=LTi(j, i)
    LTi(j, i)=LTi(j, i)/LTi(i, i)
  ENDDO
  DO jy=i+1, ist+7
  DO jx=jx, ist+7
    LTi(jx, jy)=LTi(jx, jy)-LTi(jx, i) × LTi(i, jy)
  ENDDO
  ENDDO
  ) (20)

  [ UPDATE LTi(LTi(ist+8:k, ist:ist+7).
    SINCE  $DL^T$  IS INCLUDED IN THE UPPER TRIANGLE OF
    LTi(LTi(ist:ist+7, ist:ist+7), UPDATE  $(PL^T)^{-1}$  FROM THE RIGHT. ]

  ELSE
    call LDL(LTi, k, iblks, ist, nwid/2)

    COPY  $DL^T$  TO
     $-LTi(ist:ist+nwid/2-1, ist+nwid/2:ist+nwid-1)$ .
    (D IS AN OBJECT ELEMENT OF LTi(ist:ist+nwid/2-1, ist:ist+nwid/2-1)
    AND L IS
    LTi(ist+nwid/2:ist+nwid-1, ist:ist+nwid/2-1),
    TRANSPOSING THIS  $L^T$ .)

    ·UPDATE LTi(ist+nwid/2:k, ist+nwid/2:ist+nwid-1).

    [ LTi(ist+nwid/2:k, ist+nwid/2:ist+nwid-1)
      =LTi(ist:ist+nwid/2:k, ist+nwid/2:ist+nwid-1)-
      LTi(ist+nwid/2:k, ist:ist+nwid-1) ×
      LTi(ist:ist+nwid/2-1, ist+nwid/2:ist+nwid-1) ]

    ·CALL LDL (LTi, k, iblks, ist+nwid/2, nwid/2)

  ENDIF

  RETURN

  END

```

FIG. 16